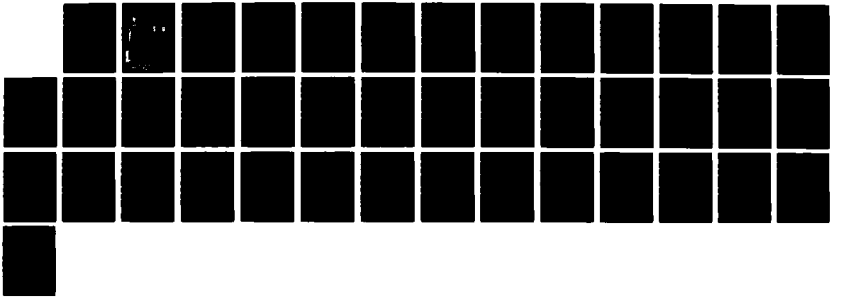
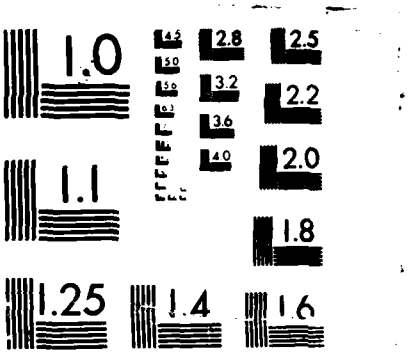
AD-A188 334 COMPUTER SIMULATION OF BOUNDED PLASMA SYSTEMS(U) 1/1
CALIFORNIA UNIV BERKELEY ELECTRONICS RESEARCH LAB
W S LAWSON 05 MAR 87 UCB/ERL-M87/14 N00014-77-C-0578
UNCLASSIFIED F/G 20/9 NL





COMPUTER SIMULATION OF BOUNDED PLASMA SYSTEMS

by

William S. Lawson

Memorandum Number UCB/ERL M87/14

5 March 1987

Contract N00014-77-C-0578

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley, CA 94720

COMPUTER SIMULATION OF BOUNDED PLASMA SYSTEMS

by

William S. Lawson

Memorandum Number UCB/ERL M87/14

5 March 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Work supported by Department of Energy Contract Number DE-AT03-76ET53064 and Office of Naval Research Contract Number N00014-77-C-0578.

COMPUTER SIMULATION OF BOUNDED PLASMA SYSTEMS

William S. Lawson

Abstract

The physical and numerical problems of kinetic simulation of a bounded electrostatic plasma system in one planar dimension are examined, and solutions to them are presented. These problems include particle absorption, reflection and emission at boundaries, the solution of Poisson's equation under non-periodic boundary conditions, and the treatment of an external circuit connecting the boundaries. Some comments are also made regarding the problems of higher dimensions. The methods which are described here are implemented in a code named PDW1, which is available from Professor C. K. Birdsall, Plasma Theory and Simulation Group, Cory Hall, University of California, Berkeley, CA 94720.

Introduction

In recent years, many plasma simulation problems have become of interest which are not spatially periodic. These problems include (*i*) devices, such as magnetic mirrors [1], Q-machines [2], and double-plasma machines, (*ii*) spatial effects such as wave damping and mode conversion, (*iii*) boundaries between different plasmas as in double layers, and (*iv*) theory problems such as the Pierce diode [3]. Over the past twenty years, many non-periodic simulations have been performed (Birdsall and Bridges [4], and Burger [5,6] are some early examples), but no systematic account of the problems involved in these simulations has been published. The purpose of this article is to begin to fill this void.

The methods described here were implemented in the computer code PDW1, and have been used successfully since 1983 by members of the Berkeley Plasma Theory and Simulation group (some examples are [7-10]). (PDW is an acronym for our 1983 Plasma Device Workshop.) The code is documented, and set up to run on the CRAY-1 computers of the National Magnetic Fusion Energy Computer Center. The code and documentation is available from the Plasma Theory and Simulation group at the University of California, Berkeley [11]. Some of the work to be presented here has already been published as part of Chapter 16 of Birdsall and Langdon [12]. Most of the

points are expanded upon here, but some points are more fully explained in [12] (particle injection, for example). Chapter 15 of [12] (discussing the work done on Langdon's ZOHAR code) is also very relevant to the work presented here. (Reference [12] is a good reference for particle simulation in general.)

The incorporation of boundaries into the simulation of a one-dimensional plasma introduces new considerations of both physical and numerical origin. These considerations may be grouped into three broad categories: the interaction of particles with the boundaries, the solution of Poisson's equation with non-periodic boundary conditions, and the inclusion of an external circuit connecting the bounding planes. The external circuit has received almost no attention in the past, but is a real part of any physical model. Even an open circuit is non-trivial, as the boundaries themselves form a capacitor. A series RLC circuit with both AC and DC voltage sources (also in series) is considered here. More complicated circuits are, of course, possible, but will not be discussed.

Each of these categories present problems which are not difficult to solve, but do require some care. This article will explore these problems category by category, and present solutions.

Model

Before considering the problems of simulating a one-dimensional plasma, it is worth drawing attention to some of the features of the one-dimensional model itself. First of all, it is important to note that the electric field of a particle (which represents a charge sheet in three dimensions) does not fall off with distance, but rather is constant. One consequence of this is that it is impossible to define a unique ground potential at infinity as is possible in three dimensions. The point chosen to be at zero potential is completely arbitrary. It may be designated in a circuit diagram as a ground symbol, but it should be remembered that there really is no ground for charge to flow to or from.

It might seem as if such an artificial model would be of little use in describing reality, but in any situation in which the variations along two axes is thought to be unimportant, the one-dimensional model is applicable. One dimensional periodic simulations have been used successfully for almost thirty years on wave problems, and non-periodic simulations show equal promise. Nor are one-dimensional simulations limited to one velocity dimension. Only the *variation* of quantities in the perpendicular directions is required to be zero. Three velocity dimensions and a constant magnetic field in an arbitrary direction have already been implemented in PDW1, and fully electromagnetic 1-d simulations have been implemented elsewhere.

Because electric field of a particle in one dimension does not fall off with distance, there is no functional difference between a boundary between two regions of a plasma and a boundary between a plasma and a conducting wall (or between a plasma and anything else) aside from the particle emission from the boundary. This is very different from the situations in two and three dimensions. In one dimension, any information a particle carries beyond a boundary is of no interest regardless of the nature of the region beyond the boundary because the electric field due to the particle will be the same whether it is located on that boundary or infinitely far beyond it. To illustrate, in Fig. 1, diagrams (a) and (b) have the same solution for the region $0 < x < L$ *if and only if* the particles which enter the simulation region from the boundaries are the same. Put more simply, particles which are outside the simulation region have no influence on the dynamics of the simulation region aside from their net charge.

According to Gauss' law,

$$E(l) - E(0) = \frac{1}{\epsilon_0} \int_0^l \rho(x)\, dx$$

so surface charge densities can be defined at the left and right hand boundaries as

$$\sigma_L = \epsilon_0 E(0)$$

and

$$\sigma_R = -\epsilon_0 E(l)$$

so that

$$\int_0^l \rho(x)\, dx + \sigma_L + \sigma_R = 0 \tag{1}$$

i.e., the total charge of the system including the boundaries is zero. This is only a definition. If the boundaries represent conducting walls, then this surface charge will be the surface charge which is physically on the walls (see Fig. 2). If the boundaries represent only an end to the region of interest, then this surface charge will represent charges *beyond* the boundary.

The electrostatic energy within the system is

$$W_E = \int_0^l \frac{\epsilon_0}{2} E^2\, dx$$

Integrating this by parts (using $E = -d\phi/dx$ and $\rho = \epsilon_0 dE/dx$)

$$\begin{aligned} W_E &= \int_0^l \frac{1}{2}\rho\phi\, dx + \frac{\epsilon_0}{2} E(0)\phi_L - \frac{\epsilon_0}{2} E(l)\phi_R \\ &= \int_0^l \frac{1}{2}\rho\phi\, dx + \frac{1}{2}\sigma_L\phi_L + \frac{1}{2}\sigma_R\phi_R \end{aligned} \tag{2}$$

This result is independent of the reference potential, thanks to Equation 1, so it is convenient to choose ϕ_L as zero for the entire simulation (in a sense choosing the left boundary as ground potential). This choice means that σ_L need not be known to calculate the electrostatic energy (which is an essential diagnostic). Since σ_L is not needed for the dynamical equations either, (and can be calculated at any time from Equation 1) σ_L can be eliminated as a dynamical variable in a simulation (this is done in PDW1).

When the boundaries represent conducting walls, the surface charge density represents a real surface charge density. Equation 2 is correct, though, even when the boundaries do *not* represent conducting walls, because W_E represents only the energy contained in the region between the boundaries.

The circuit model with all of its dynamic variables can now be diagrammed (see Fig. 3). The background current, shown as J_{BACK} in Fig. 3, has not been mentioned. It can represent either current due to a massive background species, or a parallel circuit containing a constant current source.

Boundary-Particle Interactions

(a) Charge Accumulation at Boundaries

In particle simulations, the electric field and potential are usually found on a spatial grid using a finite difference approximation of Poisson's equation. To do this, the charge density must be known on the same grid, and the algorithm for assigning the charges of the particles to the grid is known as charge accumulation. Many such algorithms exist, but the most commonly used algorithm is called linear weighting (also sometimes CIC, Cloud in Cell, or PIC, Particle in Cell). In linear weighting, a particle's full charge is assigned to a grid point if it is exactly on it, and if it is not, the fraction of its charge which is assigned to the grid point varies linearly with its distance from the grid point, reaching zero when the particle sits exactly on the next grid point. This weighting scheme can be represented graphically (see Fig. 4). Linear weighting is popular because it has the advantages of being simple (and therefore fast), and continuous, which reduces noise.

The problem now is how to cope with a boundary. The method implemented in PDW1 requires both a grid point *at* the boundary, representing the plasma charge density *adjacent* to the boundary, and a surface charge *on* the boundary (see Fig. 5). The method is diagrammed in Fig. 6. Note that

when a particle's position shifts from outside the boundary to inside the boundary, its charge jumps suddenly from being part of the charge density *at* the boundary to being part of the surface charge *on* the boundary. It might seem that this sudden jump would cause some unwanted noise, but it will be shown in the section on solving Poisson's equation that the noise is limited to the first grid cell, and has virtually no effect. It is important to remember with this algorithm that the charge accumulated at the grid point at the boundary is only half of that which would be accumulated at a grid point which is not at a boundary if the *physical charge density* were the same at the two points. This can readily be seen from Fig. 6, as the triangle representing the charge collection of the grid point at the boundary has only half the area of the other grid points. In this sense, the grid cell adjacent to the boundary is only half as wide as a normal grid cell, and this must be compensated for. Thus for diagnostic purposes, if for nothing else, it is necessary to double the charge collected at the boundary to obtain the physical charge density. (Birdsall and Langdon [12] actually *define* ρ_0 to be half of $\rho(0)$. This is natural from the standpoint of a computer algorithm, but may cause some confusion.)

This "hard" boundary is more difficult to implement when weighting schemes other than the linear one are used, as they may assign charge to grid points which are more than a single grid spacing away from the position of a particle. The finite difference equations for the electric field and potential must be carefully designed to prevent the noise of absorption from affecting the potential throughout the plasma (see the section on the solution of Poisson's equation).

Other schemes which fit into the linear weighting method are "soft" boundaries. In these schemes particles enter the boundary gradually, *i.e.*, the charge of a particle near the boundary is assigned partly to the boundary and partly to the grid. Two such schemes are diagrammed in Fig. 7. The first of these simply merges the plasma charge density at the boundary with the charge on the boundary. This method is numerically equivalent to the previous "hard" boundary method, but does not give the "real" charge on the boundaries, which may be an inconvenience. The second method is more faithful to the spirit of linear weighting. The charge density at the boundary exists, but particles are absorbed into the boundary slowly, and thus create no noise. The problem with "soft" boundary methods is that a particle which has been partially absorbed can turn around and leave the neighborhood of the boundary, taking away charge which has supposedly already been absorbed. This is clearly unphysical, although it may not always be a problem. If it is known beforehand that particles will always be entering the boundary at a large velocity, or that the particles which are

almost reflected at the boundary will play no role in the dynamics, then these methods are quite usable, though they are unlikely to be necessary. The methods described in the rest of this article should apply just as well to both types of boundary.

(b) Particle Absorption

Another problem created by the incorporation of boundaries into the model is the removal of particles which have left the simulation region (*i.e.*, been absorbed by a boundary). Conceptually there is scarcely a problem: once a particle is absorbed, it can be removed from the simulation. This removal, however, creates several algorithmic problems. First, it means that the number of particles in the simulation will vary with time. This requires some foreknowledge of the maximum number of particles which will exist in the simulation, since a fixed amount of computer memory is usually allocated to the particle array. (Some compilers allow more memory to be allocated when it is needed, but this can waste valuable computation time.) Second, the removal of particles which have left the system can be an expensive process computationally, so an efficient algorithm is needed. In the computer, particles are assigned to memory locations in an array, and when particles are removed, they will generally leave holes in the array. The algorithm which removes particles should also repack the particle array so that the array is no longer than the number of particles which are still inside the simulation region, since a compact array makes the rest of the simulation more efficient.

The algorithm which is implemented in PDW1 uses two pointers. The first pointer starts at the first particle in the array, and checks to see whether the particle is still inside the system. If it is, then the pointer moves on to the next particle in the array. The other pointer marks the last active particle in the array. When a particle is found which is not inside the system, the last active particle is moved into its place in the array, and the second pointer is moved back to the next-to-last particle (which now *is* the last particle in the array). The process of checking whether particles are still within the simulation region then resumes, beginning with the particle whose location in memory has just been changed. Some care must be taken to be sure that when the two pointers come together, no particles are lost.

Since every particle must be tested, this algorithm is not vectorizable, and so the removal and repacking of particles is very time consuming on a vector machine (such as the CRAY-1). Fortunately, there is no reason why this process must occur every time step. Particles which are

beyond the boundaries can have their charge added to the charge of the boundary *temporarily* when the charges of the particles are being weighted to the grid to obtain the charge density. Then when they are removed their charge can be permanently assigned to the boundary itself. This has the disadvantage that it complicates the bookkeeping necessary to account for all the charges. Errors in such bookkeeping will have drastic effects.

(c) Particle Reflection

Other problems arise in the treatment of particles which are reflected from a boundary. For simplicity, let us consider reflection at the left-hand boundary. First, if particles are simply reflected, meaning $x \leftarrow -x$ and $v_x \leftarrow -v_x$, and the leapfrog method (which is almost universal) is being used, a first-order error occurs in the reflected velocity. (The notation $x \leftarrow -x$ means that the value of the x coordinate is to be replaced with the negative of the value it would have had if it were not being reflected. This might seem backward, but the notation is intended to mimic a computer language assignment statement.) Since the cumulative error of the leapfrog method is second order, it is worth while, if not essential, to correct this first-order error.

This correction can be derived from energy conservation or from considering the net effect the electric field near the boundary should have both before and after reflection, or by a careful analysis of the force on a particle, as in the derivation below. The error can be understood by considering the difference between a physical particle being reflected and a simulation particle being reflected. The velocity of a physical particle, and therefore the rate of work done on it $v \cdot E$, reverses the instant it is reflected. A simulation particle, on the other hand, is accelerated for a full time step, and is *then* reflected. Thus the energy change is all of one sign, and a first-order error results. The details of this picture are complicated by the temporal offset between x and v_x (x is known at integral time steps and v_x is known at half-integral time steps).

The derivation will consist of five steps: an acceleration of half a time step to synchronize the particle velocity with the particle position, advancement of the position and velocity to the time of reflection, reflection, advancement to the end of the time step, and a deceleration of half a time step to put the velocity back to the correct time for the leapfrog scheme. For simplicity, reflection from a left hand wall at $x = 0$ will be considered for the derivation. The position and velocity are known *after* the particle has passed through the wall, so these are the natural variables to define as x, and v_x. Let t_0 be the time at which x is known, so that $t_{-1/2}$ is when v_x is known. The reflection occurs

somewhere between t_{-1} and t_0 (see Fig. 8). The electric field can be assumed to be uniform in space and time, since it should vary only slightly over the distance and time a particle travels in a single time step.

At $t = t_{-1}$ the velocity is

$$v_{-1} = v_s - \frac{qE}{m}\frac{\Delta t}{2}$$

(remember that $v_s = v_{-1/2}$). At the moment of reflection, the velocity has been acclerated for a time $\Delta t - x/v_s$ to first order, so the velocity just before reflection is

$$\begin{aligned} v^- &= v_{-1} + \frac{qE}{m}\left(\Delta t - \frac{x}{v_s}\right) \\ &= v_s + \frac{qE}{m}\left(\frac{\Delta t}{2} - \frac{x}{v_s}\right) \end{aligned}$$

Reflection just reverses this velocity, so

$$v^+ = -v_s + \frac{qE}{m}\left(\frac{x}{v_s} - \frac{\Delta t}{2}\right)$$

The time interval from the time of reflection to t_0 is x/v_s, so

$$\begin{aligned} v_0 &= v^+ + \frac{qE}{m}\frac{x}{v_s} \\ &= -v_s + \frac{qE}{m}\left(2\frac{x}{v_s} - \frac{\Delta t}{2}\right) \end{aligned}$$

The desired velocity is $v_{-1/2}$ *after* reflection, which is

$$\begin{aligned} v_s^* &= v_0 - \frac{qE}{m}\frac{\Delta t}{2} \\ &= -v_s + \frac{qE}{m}\left(2\frac{x}{v_s} - \Delta t\right) \end{aligned}$$

This result can be rewritten so that it will apply to any boundary, so that the rule for reflection is now

$$v_s \leftarrow -v_s + \frac{qE}{m}\left(2\frac{d}{|v_s|} - \Delta t\right) \tag{3}$$

where d is the distance between the particle and the boundary after reflection (d is always positive).

Since it is to be expected that d for different particles will be uniformly distributed between 0 and $|v_s|\Delta t$, the correction term (in parentheses) will be zero on the average. Thus if the reflected

velocity is not corrected, the effect will not be catastrophic, but will rather cause some diffusion in velocity space. The root mean square velocity change of an ensemble of particles will be

$$\sigma_v = \frac{qE}{m}\frac{\Delta t}{\sqrt{3}} \tag{4}$$

independent of velocity. Depending on the distribution, this diffusion will most likely cause some heating (the magnitude of this effect is calculable given the distribution), but since reflection occurs infrequently for any given particle, the effect may well be unimportant, particularly if the electric field is small.

Another difficulty arising from reflection is physical. When a magnetic field is present, the reflection algorithm must depend on the physical situation the reflection is meant to be modeling. If the reflection represents an actual physical reflection, as at a wall, then $x \leftarrow -x$ and $v_x \leftarrow -v_x$ (with correction) is proper. If, however, the reflection is meant to model a symmetry plane, then an inversion ($x \leftarrow -x$, $v_x \leftarrow -v_x$, $v_y \leftarrow -v_y$ and $v_z \leftarrow -v_z$) is required to preserve the direction of the magnetic field through the reflection. This is a subtle physical point, but important; simple reflection does not conserve a particle's magnetic moment, but inversion does (see Section 14-7 of [12]).

(d) Particle Emission

The last boundary-particle problem is particle emission (or injection). This problem is closely related to that of initial loading of a simulation, and this discussion assumes some knowledge of loading techniques. (See reference [12] for a thorough discussion of both initial loading and injection techniques.) The fundamental difference between loading initially ($t = 0$) and injecting during ($t > 0$) a simulation lies in the region of phase space which is to be filled with particles. For the initial load, the entire simulation phase space is usually filled, whereas for injection, only a wedge-shaped region of phase space need be filled (see Fig. 9). The random variables for the initial load are x and v, whereas the random variables for the injection are t and v. Both distributions are typically uniform in one variable (x and t respectively), and non-uniform in v. The uniformity in the one variable is not as important as the statistical independence of the variables — lack of independence makes the loading and injection problems much more difficult. The loading distribution function must be known. Let it be denoted as $f(v)$ (maintaining the assumption that $f(v)$ is independent of x and t). In terms of $f(v)$ the injection (flux) distribution is just $vf(v)$ (suitably normalized), thus the two problems can be solved by the same methods.

The problem of injection has complications which the loading problem does not; for instance (*i*) the charge of each particle injected must be subtracted from the charge of the boundary, and (*ii*) the number of particles injected each time step must be chosen to average to the desired number without resorting to fractional particles.

Another injection problem is the temporal offset between position and velocity. In the standard leap-frog integration scheme, the positions are known at integral time steps, and the velocities are known at half-integral time steps. It is important that the injected particles (which are injected on the basis that position and velocity are known at the same time) be given an acceleration (but no displacement) of a fraction of a time step (which depends on the time of emission, and may be negative) in order to synchronize them with the rest of the particles (see [12] for a discussion of the leapfrog scheme).

A subtle and insidious injection effect has recently been discovered and explained (this work is being prepared for publication in the Journal of Computational Physics in the near future). The effect is the artificial cooling of a trapped population of electrons. The explanation is a combination of simulation and physical effects which has not been previously observed due to the rarity of long bounded simulations with trapped particles. The effect relies on the thermodynamic properties of what is known to simulators as "quiet" injection.

In most simulations, fluctuations are a problem, and so to minimize them particles are put into the system with artificial regularity in order to reduce the noise produced by the particles (see [12] for a discussion of quiet loading techniques). This noise, however, includes the fluctuations which must occur in a plasma which is in thermodynamic equilibrium with the waves it supports. (It is helpful to use the classical idea of a wave field which is in thermodynamic contact with the particles.) In a particle simulation with quiet injection of particles, the wave fluctuation level is much smaller than that which would occur in thermodynamic equilibrium, and so the wave field represents a cold system in contact with the plasma.

The cooling of the trapped electrons can be understood from this new viewpoint. Because of the cold wave field, the particles will all be cooling somewhat. Only the trapped particles, however, stay in the system long enough to cool noticeably. One might expect the wave field to warm up (increasing the fluctuation level), but the system is bounded, so the fluctuations can leave the system, primarily carried by the particles, only to be replaced by the constant injection of quiet particles. Put more simply, the trapped particles radiate, that radiation leaves the system, and there is no

external source of fluctuations to warm the particles (as there would be in a state of thermodynamic equilibrium). This effect is not observed in periodic simulations because the fluctuations generated cannot leave the system, and the energy in the fluctuations is small compared with the thermal energy of the particles. A more detailed explanation, and the simulation evidence for it, will be published separately soon.

Other boundary effects are possible, and for the most part are understandable in terms of the problems so far considered. For example, L. A. Schwager has succeeded in including secondary emission of electrons, which is dealt with as a combination of absorption and injection, and P. G. Gray has included ionization over a region of the simulation. (None of this work is yet published.)

Solution of Poisson's Equation

There are two differences in the solution of Poisson's equation in a bounded 1-d system as opposed to a periodic one. These differences are: allowing for the net non-neutrality of the system, and satisfying the boundary conditions. The boundary conditions in a periodic system are simply that the potential and electric field be equal at each end of the simulation region. (It is possible to add an overall electric field, which would make the potential difference across the system non-zero, but the periodicity of the electric field is absolute.) The periodicity of the electric field requires that

$$\frac{1}{\epsilon_0}\int_0^l \rho\, dx = E(l) - E(0) = 0$$

i.e., the system must be net neutral. This constraint arises because the boundary conditions go beyond the conditions necessary for uniqueness of the solution.

In a bounded (non-periodic) system, the boundary equations are not overspecified, and the net charge is not necessarily zero. The boundary conditions are one of the familiar types: Neumann, Dirichlet, or mixed. These are, respectively, a given electric field at one end (and the potential at some point), a given potential difference between the boundaries (and the potential at some point), or some linear relation between the electric field at one end and the potential difference between the boundaries (and the potential at some point). Fortunately, since the homogeneous solution of Poisson's equation is so simple (a uniform electric field), it is easy to solve the system by first using almost any boundary conditions, then adding in the homogeneous solution to satisfy the desired boundary conditions.

The finite diference equation most commonly used in one dimension for Poisson's equation is

$$\frac{\phi_{n+1} - 2\phi_n + \phi_{n-1}}{\Delta x^2} = -\frac{\rho_n}{\epsilon_0} \tag{5}$$

with the auxiliary definition of the electric field

$$E_n = \frac{\phi_{n-1} - \phi_{n+1}}{2\Delta x} \tag{6}$$

These E_n's satisfy

$$\frac{E_{n+1} - E_n}{\Delta x} = \frac{(\rho_{n+1} + \rho_n)}{2\epsilon_0} \tag{7}$$

These values of E_n are used to accelerate particles. Another set of E's can be defined, which are more useful in understanding the difference equations. They are

$$E_{n+1/2} = -\frac{\phi_{n+1} - \phi_n}{\Delta x} \tag{8}$$

These E values satisfy the simpler equation

$$\frac{E_{n+1/2} - E_{n-1/2}}{\Delta x} = \frac{\rho_n}{\epsilon_0} \tag{9}$$

The values of E with integer indices can be expressed in terms of the half-integer values as

$$E_n = \frac{E_{n+1/2} + E_{n-1/2}}{2} \tag{10}$$

The half-integer values are not used to accelerate particles (although they could be), and are not even equal to the field used to accelerate a particle which is half way between two grid points. The interest here in these half-integral field values is theoretical.

The difference equation version of Equation 2 can now be used to determine how the boundaries must be handled. Starting from the integral of $\rho\phi$,

$$\begin{aligned}
W_E &= \frac{1}{2}\left[\sum_{n=1}^{N-1} \rho_n\phi_n\,\Delta x + \left(\frac{\rho_0}{2}\phi_0 + \frac{\rho_N}{2}\phi_N\right)\Delta x\right] + \frac{1}{2}(\sigma_L\phi_0 + \sigma_R\phi_N) \\
&= \frac{\epsilon_0}{2}\sum_{n=1}^{N-1}(E_{n+1/2} - E_{n-1/2})\phi_n + \frac{1}{2}\left(\frac{\rho_0}{2}\phi_0 + \frac{\rho_N}{2}\phi_N\right)\Delta x + \frac{1}{2}(\sigma_L\phi_0 + \sigma_R\phi_N) \\
&= \frac{\epsilon_0}{2}\sum_{n=1}^{N-2} E_{n+1/2}(\phi_n - \phi_{n+1}) + \frac{1}{2}\left[\epsilon_0 E_{1/2}\phi_1 + \frac{\rho_0}{2}\phi_0\Delta x\right. \\
&\quad \left. + \epsilon_0 E_{N-1/2}\phi_{N-1} + \frac{\rho_N}{2}\phi_N\Delta x + (\sigma_L\phi_0 + \sigma_R\phi_N)\right] \\
&= \frac{\epsilon_0}{2}\sum_{n=0}^{N-1} E_{n+1/2}^2\Delta x + \frac{1}{2}\left(\sigma_L + \frac{\rho_0}{2}\Delta x - \epsilon_0 E_{1/2}\right)\phi_0 \\
&\quad + \frac{1}{2}\left(\sigma_R + \frac{\rho_N}{2}\Delta x + \epsilon_0 E_{N-1/2}\right)\phi_N
\end{aligned}$$

The summation is the desired integral, so that the rest of the terms should be zero for all choices of the ground potential. Since the two boundaries must be independent,

$$E_{1/2} = \frac{1}{\epsilon_0}\left(\sigma_L + \frac{1}{2}\rho_0\Delta x\right)$$
$$E_{N-1/2} = -\frac{1}{\epsilon_0}\left(\sigma_R + \frac{1}{2}\rho_N\Delta x\right) \tag{11}$$

are the necessary choices. We can now do away with the half-integer E's, and the result is two new conditions:

$$\phi_1 = \phi_0 - \frac{1}{\epsilon_0}\left(\sigma_L + \frac{\rho_0}{2}\Delta x\right)\Delta x$$
$$\phi_{N-1} = \phi_N - \frac{1}{\epsilon_0}\left(\sigma_R + \frac{\rho_N}{2}\Delta x\right)\Delta x \tag{12}$$

If the boundary condition is the Dirichlet type (fixed bias external circuit), then Equations 12 give the surface charges. If the boundary condition is the Neumann type (open external circuit), then one of the equations will *be* the boundary condition, and the other will give the other surface charge. If the external circuit is anything but open or fixed bias, the circuit equation will impose a time-varying mixed boundary condition which will supply the last condition for uniquely specifying the solution.

It can now be seen why the absorption of particles does not create noise throughout the system. When a particle is absorbed, its charge moves from being counted as part of $\rho_N/2$ to being counted as part of σ_R. Thus ϕ_{N-1} is not affected by the noise of absorption. The noise is limitted to σ_L and σ_R, and therefore to E_0 and E_N. Some people (A. B. Langdon, for example) prefer to use $E_{1/2}$ and $E_{N-1/2}$ instead of E_0 and E_N to accelerate particles, avoiding the noise in favor of a small systematic error. This use of $E_{1/2}$ and $E_{N-1/2}$ for E_0 and E_N is *only* for the acceleration of particles.

Once Poisson's equation has been turned into a finite difference equation, and the boundary conditions have been similarly converted, its solution is unique and therefore independent of the method used to solve it; however, some methods of solution offer advantages over others. I will describe three methods of solving the finite difference Poisson equation. The first two are based on the Fast Fourier Transform (FFT), and the third is a direct "marching" method. The third method is the one which is implemented in PDW1, but the others have been used in the past.

The preferred method for solving Poisson's equation in *periodic* simulations is the Fast Fourier Transform. The FFT has three important advantages: first, it is naturally periodic, second, it is

vectorizable on CRAY class computers, which allows an appreciable increase in speed, and third, in a periodic system, the different Fourier modes often have physical meaning (wave modes), and the FFT produces modal information as a by-product.

An FFT can be used in solving the non-periodic case also, but some preconditioning is required, and two of the three advantages are lost. The natural periodicity of the FFT becomes a liability rather than an advantage, and the Fourier modes will rarely if ever carry any physical significance, thus making the extra time and effort of the FFT time and effort wasted. The vectorizability of most of the FFT algorithm remains, however, which may make FFT methods profitable despite the complexity of the algorithm.

Because of the inherently periodic nature of the FFT, a non-periodic system must be manipulated into a periodic one if the FFT is to be used to solve it. The first method does this doubling the length of the system (for purposes of the field solution only), and defining the charge density over the newly introduced half of the system so as to make the entire system periodic, *i.e.*,

$$\rho_{2N-n} = -\rho_n \tag{13}$$

Note that this periodicity also forces ρ_N and $\rho_0 (= \rho_{2N})$ to be zero, which are not the given values for these variables. All the Equations 5 are satisfied, however, and since the points at $n = 0$ and $n = N$ do not satisfy the Equations 12 anyway, the further incorrect treatment of these points is of no consequence. Once the FFT solution has been obtained, the uniform field (homogeneous) solution ($\phi_n = E_0 \cdot n\Delta x$, E_0 arbitrary) can be added to the FFT solution with E_0 chosen so that the sum of the FFT solution and the uniform field solution *will* satisfy Equations 12. This method has the obvious drawback of using twice as much space and time as a similar periodic system. Since the solution of Poisson's equation often requires very little time and space relative to the particle integration, this may not be an important drawback.

Another FFT method is to subtract the average charge density from the full charge density, thus producing two problems: one which is net charge neutral, and one which has a uniform charge density, *i.e.*,

$$\nabla^2 \phi_1 = \bar{\rho} \tag{14}$$

$$\nabla^2 \phi_2 = \rho - \bar{\rho} \tag{15}$$

$$\phi = \phi_1 + \phi_2$$

It is important to define $\bar{\rho}$ correctly:

$$\bar{\rho} = \frac{\Delta x}{l}\left(\frac{\rho_0}{2} + \sum_{n=1}^{N-1}\rho_n + \frac{\rho_N}{2}\right) \tag{16}$$

The net neutral (ϕ_2) problem can be made periodic by simply wrapping the system around so that $n = 0$ and $n = N$ represent the same point with charge density $(\rho_0 + \rho_N)/2 - \bar{\rho}$ (to maintain the charge neutrality). This problem can then be solved using the FFT.

The uniform charge density problem can be solved algebraically since the finite difference equation and the differential equation have exactly the same solution for a uniform charge density. The solution is just

$$\phi_n = \frac{\bar{\rho}}{2\epsilon_0}(n\Delta x)^2 \tag{17}$$

The sum of the uniform and periodic solutions will satisfy everything but Equations 12 so, as in the previous FFT method, a uniform electric field must be added in order to satisfy them.

Before describing the third method of solution, it is important to point out one common method of FFT solution which does *not* work. The FFT of Equation 5 is

$$4\sin^2\frac{k\Delta x}{2}\cdot\tilde{\phi}_k = \frac{\Delta x^2}{\epsilon_0}\tilde{\rho}_k \tag{18}$$

but some simulators prefer a "non-localized" solver which uses the equation as derived from the continuous Fourier transform:

$$k^2\cdot\tilde{\phi}_k = \frac{1}{\epsilon_0}\tilde{\rho}_k \tag{19}$$

The dangers of this method have long been known (see Appendix E of [12]), but it does not usually produce serious errors for periodic systems. When the FFT is used to solve a system in which the charge density comes from a non-periodic system, however, the sudden change in the charge density at the boundaries produce large values for the high wave number components, and the solution via (19) is very different from the solution via (18). The inaccuracy will be worst near the boundaries (the site of the sudden change), and this is where the boundary conditions are to be applied. Thus severe error can arise from the inappropriate use of the FFT method of solving Poisson's equation.

The alternative to the FFT methods of solution, is a direct solution which moves from one end of the simulation region to the other. This method is not vectorizable, but in most one-dimensional gridded particle simulations, the solution of Poisson's equation still accounts for a small fraction of the computation time.

The method begins at one boundary of the simulation region, for example the right hand boundary in PDW1, and assumes some value for the surface charge and some value for the potential at that boundary. The appropriate one of Equations 12 is then solved for the first *interior* value of the potential, for example in PDW1,

$$\phi_{N-1} = \phi_N - \frac{1}{\epsilon_0}\left(\sigma_R + \frac{\rho_0}{2}\Delta x\right)\Delta x \tag{20}$$

Now two adjacent values of the potential are known. Equation 5 can be rewritten now to use any two adjacent values of the potential to find the unknown value next to them, for example,

$$\phi_{n-1} = 2\phi_n - \phi_{n+1} - \frac{\rho_n}{\epsilon_0}\Delta x^2 \tag{21}$$

Thus it is possible to simply march across the system, ending up with a value for ϕ at the opposite boundary. This solution can then be adjusted by adding a uniform field solution such that Equations 12 are both properly satisfied. In PDW1, a constant is first subtracted from all values of ϕ so that ϕ_0 will be zero. The boundary condition, as determined by the external circuit, is then applied to find the necessary uniform field to be added to satisfy Equations 12.

External Circuit

The external circuit is the most easily overlooked feature of non-periodic plasma simulations. Often, an open circuit (*i.e.*, no charge is transported from one boundary to the other except as plasma particles) is desired, and it might be argued that this is no circuit at all. The charge which passes the boundaries must still be carefully accounted for, however, and in fact, the two boundaries behave in a sense like a capacitor. Thinking in terms of an external circuit, even when the circuit is open, is a good way of ensuring that charge is not transported from one boundary to the other in a way which is inconsistent with the physical model.

An external circuit has its own intrinsic time scales which have nothing to do with the plasma time scales. For example, an RC circuit would have an RC decay time constant, and an LC circuit would have a frequency of oscillation of $1/\sqrt{LC}$. The simulation time step must be small enough to resolve these time scales, or the circuit simulation will produce inaccurate results regardless of the plasma parameters.

The external circuit can be arbitrarily complex, and circuit simulation is a large field in its own right. The external circuit chosen for consideration here is a series RLC circuit (see Fig. 3).

In addition to the RLC elements, the circuit has both AC and DC voltage sources. The circuit equations (including the field equation) are:

$$L\ddot{Q} + R\dot{Q} + \frac{Q}{C} = V(t) - V_{ext}(t) \tag{22}$$

$$A\dot{\sigma} = -\dot{Q} + AJ_{plasma}(x = l) \tag{23}$$

and

$$\nabla^2\phi(x,t) = -\frac{\rho(x,t)}{\epsilon_0} \tag{24}$$

where the boundary conditions on the electric potential are

$$\phi(0,t) = 0$$

$$\left.\frac{d\phi}{dx}\right|_{x=l} = -\frac{\sigma}{\epsilon_0} \tag{25}$$

and

$$V(t) = \phi(x = l, t)$$

in accordance with Kirchoff's voltage law. The area of the boundary is A (this is introduced for dimensional reasons; it is the cross-section of the plasma), R, L, and C represent the external circuit resistance, inductance, and capacitance, and l is the length of the plasma system. J_{plasma} is the net current density from the plasma at the right hand boundary, *i.e.*. the absorbed current less the emitted current.

The mathematical character of these equations depends on the order of the primary circuit equation (22). If $L \neq 0$ then the circuit equation is of second order, which is the same order as the equation of motion for the particles of the plasma. If $L = 0$ but $R \neq 0$ then the circuit equation is of first order, and becomes a time-dependent mixed boundary condition on the field equation (24). If $L = R = 0$, then the circuit equation becomes a simple mixed boundary condition on the field equation. Let us consider each of these three cases in detail.

When $L \neq 0$, the circuit equation is of the same order as the particle equations, and can be solved by the same type of leapfrog scheme, with the addition of the resistive term. (Other schemes are, of course, possible, but this is the simplest one which is second order accurate in time, and it meshes nicely with the particle mover.) Just as the position and acceleration of a particle is known at integral time steps and the velocity at half-integral time steps, so the charge and voltage are

known at integral time steps, and the current ($I = \dot{Q}$) is known at half-integral time steps. The difference equations for the circuit equation (22) are:

$$I_{n+1/2} = (Q_{n+1} - Q_n)/\Delta t \tag{26}$$

and

$$L\frac{(I_{n+1/2} - I_{n-1/2})}{\Delta t} + R\frac{(I_{n+1/2} + I_{n-1/2})}{2} + \frac{Q_n}{C} = V_n - V_{ext}(t_n) \tag{27}$$

These difference equations yield second order accuracy in time. Equation 27 can be solved for $I_{n+1/2}$, and Equation 26 can be solved for Q_{n+1}, thus advancing the circuit. The surface charge σ is advanced by subtracting from it the amount which was added to Q (this is only the circuit contribution to the change in σ — the plasma contribution must also be taken into account).

The way in which the advancement of the circuit is melded with the advancement of the plasma is diagrammed in Fig. 10. From a known voltage (implying a known potential within the plasma), the particles and the circuit can be advanced *simultaneously*. This advancement yields a new charge density ρ, and a new surface charge σ (both the plasma and the circuit have altered σ), which allow the solution of the field equation (24) producing the new potential and voltage. Numerically, the solution of the circuit equation is equivalent to the solution of a single particle equation, and should require about as much time.

The initial conditions for the circuit equations require some care. It is natural to assign an initial value for the capacitor charge Q and an initial value for the current I, since these are the standard initial conditions for Equation 22. One more variable must be given an initial value in order to completely specify the initial conditions. Two logical candidates for this variable are V and σ. The choice between them is one of taste, as either one can be derived from the other. My personal preference is for V as it is more immediately relevant to the dynamical equations, is an experimentally measurable quantity, and is generally more often of interest. Since the circuit difference equation is a leap-frog scheme, it is also necessary to introduce the initial half time step offset between the quantities which are to be calculated at integral time steps (Q and V) and those which are to be calculated at half-integral time steps (I). This is done in essentially the same way that it was done initially for the particles (again, see [12] for a discussion):

$$I_{-1/2} = I_0 - \frac{1}{L}\left[V_0 - V_{ext}(t_0) - RI_0 - \frac{Q_0}{C}\right]\frac{\Delta t}{2} \tag{28}$$

When $L = 0$ (but $R \neq 0$), the circuit equation is only of first order. The leapfrog scheme cannot be used to advance the circuit in this case because it is numerically unstable for small values of L (including zero). The reason for this is related to the singularity of the limit $L \to 0$. The circuit equation has two (complex) frequencies associated with it for $L \neq 0$, but only one for $L = 0$. As L approaches zero, the solutions of Equation 22 (with $V = 0$) have two decay rates which approach R/L and $1/RC$. Physically, the R/L decay rate becomes infinite, meaning that the solution corresponding to it goes to zero instantaneously. The finite difference analogue to the decay factor $\exp(R\Delta t/L$ is

$$\alpha = \frac{1 - R\Delta t/2L}{1 + R\Delta t/2L} \tag{29}$$

which is quite different from the exponential decay factor desired if $R\Delta t/L \geq 1$. For very small L, α approaches -1. A simple way of dealing with this circuit is to use a backward Euler method (which is only first-order accurate in Δt). This might seem unsatisfying in view of the second-order accuracy of the previous case, but the method has some virtues. The new circuit difference equation is

$$\frac{R}{\Delta t}(Q_{n+1} - Q_n) + \frac{Q_{n+1}}{C} = V_{n+1} - V_{ext}(t_{n+1}) \tag{30}$$

Note that the equation requires the voltage at time step $n + 1$. This method requires a different approach than the $L \neq 0$ case. Equation (30) is useful because the capacitance between the boundaries of the simulation region (with the plasma locked in place) is known to be $C_0 = A\epsilon_0/l$. Thus,

$$V_{n+1} - V'_{n+1} = -\frac{1}{C_0}(Q_{n+1} - Q_n) \tag{31}$$

where V'_{n+1} is the voltage which has been computed from the *new* particle positions, and the *old* value of Q, *i.e.*, *after* the particles have been advanced, but *without* letting the circuit advance. Equations (30) and (31) can now be combined and solved for Q_{n+1}, then V_{n+1} can be computed from either equation, thus advancing the circuit. A simple intuitive picture for this method is that the particles are advanced while the circuit is held fixed, then the circuit is relaxed, while the particles are held fixed. This scheme (which is implemented in PDW1) is diagrammed in Fig. 11.

As was mentioned, this method is only first-order accurate in time. Second-order accuracy can be achieved using the following scheme (see Appendix A for the derivation). Again, let $C_0 = \epsilon_0 A/l$ be the capacitance between the boundaries. If the plasma were not present, the circuit would be a

simple RC circuit with capacitance C' satisfying

$$\frac{1}{C'} = \frac{1}{C} + \frac{1}{C_0} \tag{32}$$

According to the exact solution of the differential circuit equation, in a single time step, the charge on the capacitor would decay by a factor of $\exp(-\Delta t/RC')$; however, due to the finite difference method, it would actually decay by a factor of $1/(1 + \Delta t/RC')$. Define R' such that

$$1 + \frac{\Delta t}{R'C'} = \exp\left(\frac{\Delta t}{RC'}\right) \tag{33}$$

then solving

$$R'\frac{Q_{n+1} - Q_n}{\Delta t} + \frac{Q_{n+1}}{C'} = \frac{V'_{n+1} + V_n}{2} + \frac{Q_n}{C_0} - \frac{V_{ext}(t_{n+1}) + V_{ext}(t_n)}{2} \tag{34}$$

for Q_{n+1} and using this to solve Equation 31 for V_{n+1} gives Q to second order. Even without this correction, the backward Euler method has the advantage that in the limits $C \to 0$ and $C \to \infty$, the numerical decay factor and the analytic decay factors are equal.

It is also possible to achieve true second-order accuracy by resorting to three or four point schemes. (Such a scheme was actually worked out by Dr. A. B. Langdon - see page 416 of [12].) These schemes introduce spurious frequencies, and require extra initial conditions, but if the scheme is chosen carefully, the spurious frequencies damp out quickly. Since the circuit advancement takes so little computer time, the only reasons for not resorting to some second-order scheme are inconvenience, and some concern that the interaction between the leapfrog particle mover and the circuit may produce unexpected results due to the differences in the algorithms. The effects of this interaction have not to my knowledge been studied. It is to be hoped that they are inconsequential.

The initial conditions for the $L = 0$ case are much the same, except that now the initial value of the current I cannot be specified, as it is determined by the initial values of Q and either σ or V. Since Equation 30 is not a leap-frog scheme, it is not necessary to create a temporal offset in the current (which is no longer a dynamical variable anyway).

The case for $L = R = 0$ represents a physical situation in which the external circuit comes to equilibrium instantaneously. Thus the concept of order of accuracy no longer applies. Inspection will show that the same numerical algorithm (Equation 30) used for the previous case ($L = 0$, $R \neq 0$) works in this case without modification. The sole difference is in the initial conditions. Now it is

possible to specify only one of Q, σ, and V at $t = 0$. If Q and σ are both specified, the system will relax in one time step to the proper state such that $Q + A\sigma$ is conserved.

Extension to Electrostatic Models of Higher Dimension

Simulation in two or three dimension is far more complicated than simulation in a single dimension (see Chapter 15 of [12]). Nevertheless, all the methods described here can be extended to two and three dimensions with the exception of the solution of Poisson's equation. The charge accumulation algorithm can be somewhat complicated by irregular conducting surfaces within the simulation region, but the concept remains simple. The charge removal algorithm is completely unchanged, and the reflection algorithm is unchanged except for having to calculate an angle of reflection when the reflecting surface is not perpendicular to an axis. The circuit remains a zero dimensional system. More complicated arrays of electrodes are possible, and more complicated circuits, but the methods for merging the plasma simulation and the circuit simulation are still valid.

The general multi-dimensional Poisson equation is rather difficult to solve for arbitrary boundary conditions, and efficient methods for its solution are a topic of active research. In non-periodic simulations in particular, the distribution of the surface charges on the boundaries must be calculated, and the homogeneous solution of Poisson's equation is no longer simple. Several methods for solving Poisson's equation in two or three dimensions are known, however, particularly in cases in which symmetry allows some simplifications. Often, in particularly simple systems (fully periodic systems, for example) the FFT can be used in all dimensions. The introduction by Hockney of the cyclic reduction technique was a breakthrough in solving systems with more complex boundary conditions, and the capacity matrix method allows the solution of problems with irregularly shaped conductors present. Hockney's review article [13] is a comprehensive exposition of these methods.

Summary

The problems created by the introduction of boundaries into a one dimensional electrostatic simulation are many, but they are all solvable through straight-forward methods. They include the problems of particles crossing boundaries, the solution of Poisson's equation with non-periodic boundary conditions, and the proper accounting of charge flow, including an external circuit. Only the solution of Poisson's equation presents new conceptual difficulties when more dimensions are

necessary, particlularly with regard to boundary conditions. These difficulties are sometimes easily solved, and sometimes extremely difficult to solve.

Acknowledgements

I gratefully acknowledge the help of all the other members of the Plasma Device Workshop: Prof. C. K. Birdsall, Dr. T. L. Crystal, Dr. S. Kuhn, Dr. M. Hudson, Dr. I. Roth, N. Otani, P. Gray, V. Thomas, L. A. Schwager, and K. Y. Kim. Although not a formal member of the workshop, Dr. A. B. Langdon of the Lawrence Livermore Laboratory also contributed greatly. I especially appreciate the help of Niels Otani in several debugging problems, and that of Lou Ann Schwager in helping to discover the need for a correction to reflected particle velocities. Special thanks are due to Drs. T. L. Crystal and S. Kuhn for forcing me to clarify, in my own mind as well as on paper, the concepts which have been presented here.

This work was supported by Department of Energy contract DE-AT03-76ET53064 and Office of Naval Research contract number N00014-77-C-0578.

Appendix A: Derivation of Equation 34

The equation to be solved is

$$R\dot{Q} + \frac{Q}{C} = V(t) - V_{ext}(t) \tag{a1}$$

The source term ($V(t)$) is not a simple source term, however, since $V(t)$ is a function of Q through the capacitance of the plasma region. To solve this equation properly, it is necessary to find a source term which depends only on the *particle* dynamics. This is done by adding Q/C_0 to both sides of Equation a1 where $C_0 = \epsilon_0 A/l$ is the vacuum capacitance of the plasma region.

$$R\dot{Q} + \frac{Q}{C} + \frac{Q}{C_0} = V(t) + \frac{Q}{C_0} - V_{ext}(t) \tag{a2}$$

The right hand side of Equation a2 is now a function only of time and the particle motions — not of the charge Q. This is because it is what the voltage would be (plus the bias) if the external circuit were open, *i.e.*, if no current flowed through the external circuit. This is easy to see, since if the charge which has flowed through the circuit into the capacitor were sent back to the boundary surface charge, the voltage would go up by Q/C_0.

This reasoning does not take into account the effect of feedback from the voltage $V(t)$ on the particle motions. If the circuit advancement is second-order accurate, though, then the forces on the particles will be accurate to second order, and if the particle trajectories are accurate to second order, then the forcing term of the circuit will be second-order accurate. Thus the second-order accuracy of the whole scheme hinges on the second-order solution of Equation a2.

When the right-hand side of Equation a2 is zero (the vacuum case), the solution for Q is a dying exponential with time constant RC' where

$$\frac{1}{C'} = \frac{1}{C} + \frac{1}{C_0}$$

The stable backward Euler finite difference method for this problem is

$$R'\frac{Q_{n+1} - Q_n}{\Delta t} + \frac{Q_{n+1}}{C'} = 0 \tag{a3}$$

(R' has been used instead of R in anticipation of the need for a correction). The solution to Equation a3 decays in one time step by a factor of $1/(1 + \Delta t/R'C')$, so that if R' is chosen so as to satisfy

$$1 + \frac{\Delta t}{R'C'} = \exp\left(\frac{\Delta t}{RC'}\right) \tag{a4}$$

then the solution to Equation a3 will be the exact solution of Equation a2.

The source term must now be added to this scheme in such a way as to retain second-order accuracy. Let the entire source term be denoted by $F(t)$, *i.e.*,

$$F(t) = V(t) + \frac{Q}{C_0} - V_{cal}(t)$$

then the correct method of introducing F into Equation a3 is

$$R' \frac{Q_{n+1} - Q_n}{\Delta t} + \frac{Q_{n+1}}{C'} = \frac{1}{2}(F_{n+1} + F_n) \tag{a5}$$

This correctness will now be shown.

To simplify the algebra, let $\tau = RC'$ and $\lambda = R'/R$, then Equation a5 becomes

$$\frac{Q_{n+1} - Q_n}{\Delta t} + \frac{Q_{n+1}}{\lambda \tau} = \frac{1}{\lambda} \frac{F_{n+1} + F_n}{2R} \tag{a6}$$

with

$$1 + \frac{\Delta t}{\lambda \tau} = \exp\left(\frac{\Delta t}{\tau}\right) \tag{a7}$$

(this is just the definition of R' rewritten in terms of λ and τ).

Solving Equation a6 for Q_{n+1} gives

$$\left(1 + \frac{\Delta t}{\lambda \tau}\right) Q_{n+1} = Q_n + \frac{1}{\lambda}\left(\frac{F_{n+1} + F_n}{2R}\right) \Delta t \tag{a8}$$

The factor multiplying Q_{n+1} is just $\exp(\Delta t/\tau)$, so

$$Q_{n+1} = \left[Q_n + \frac{\Delta t}{\lambda}\left(\frac{F_{n+1} + F_n}{2R}\right)\right] \exp\left(-\frac{\Delta t}{\tau}\right) \tag{a9}$$

Each of the quantities on the right hand side must now be expanded to second order in Δt.

The exponential is just

$$\exp\left(-\frac{\Delta t}{\tau}\right) = 1 - \frac{\Delta t}{\tau} + \frac{1}{2}\frac{\Delta t^2}{\tau^2} \tag{a10}$$

Rather than expand $1/\lambda$, it is easier to expand $\exp(-\Delta t/\tau)/\lambda$ using Equation a4:

$$\begin{aligned} \frac{1}{\lambda} \exp\left(-\frac{\Delta t}{\tau}\right) &= \frac{\tau}{\Delta t}\left[1 - \exp\left(-\frac{\Delta t}{\tau}\right)\right] \\ &= \left(1 - \frac{1}{2}\frac{\Delta t}{\tau}\right) \end{aligned} \tag{a11}$$

(Only a first-order expansion is needed here since the term containing λ is multiplied by Δt.) The source function F can also be expanded:

$$F_{n+1} = F_n + \dot{F}_n \Delta t \tag{a12}$$

(Again, only a first-order expansion is necessary.) So,

$$\frac{1}{2}(F_{n+1} + F_n) = F_n + \frac{1}{2}\dot{F}_n \Delta t \tag{a13}$$

This can be simplified by using the differential equation

$$\frac{F_n}{R} = \dot{Q}_n + \frac{Q_n}{\tau}$$

and its derivative

$$\frac{\dot{F}_n}{R} = \ddot{Q}_n + \frac{\dot{Q}_n}{\tau}$$

to get

$$\frac{F_{n+1} + F_n}{2R} = \frac{Q_n}{\tau} + \dot{Q}_n + \frac{1}{2}\dot{Q}_n \frac{\Delta t}{\tau} + \frac{1}{2}\ddot{Q}_n \Delta t \tag{a14}$$

Substituting all these expressions into Equation a9,

$$\begin{aligned} Q_{n+1} &= Q_n \left(1 - \frac{\Delta t}{\tau} + \frac{1}{2}\frac{\Delta t^2}{\tau^2}\right) \\ &\quad + \Delta t \left(1 - \frac{1}{2}\frac{\Delta t}{\tau}\right)\left(\frac{Q_n}{\tau} + \dot{Q}_n + \frac{1}{2}\frac{\Delta t}{\tau}\dot{Q}_n + \frac{1}{2}\ddot{Q}_n \Delta t\right) \\ &= Q_n + \dot{Q}_n \Delta t + \frac{1}{2}\ddot{Q}_n \Delta t^2 \end{aligned} \tag{a15}$$

to second order. This is the correct expansion for Q_{n+1} to second order, so the scheme is second-order accurate.

To derive Equation 34 it remains only to manipulate the right hand side of Equation a5 using Equation 31:

$$\begin{aligned} &\frac{1}{2}\left(V_{n+1} + \frac{Q_{n+1}}{C_0} + V_n + \frac{Q_n}{C_0}\right) \\ &\quad = \frac{1}{2}\left(V'_{n+1} + \frac{Q_n}{C_0} - \frac{Q_{n+1}}{C_0} + \frac{Q_{n+1}}{C_0} + V_n + \frac{Q_n}{C_0}\right) \\ &\quad = \frac{1}{2}(V_n + V'_{n+1}) + \frac{Q_n}{C_0} \end{aligned} \tag{a16}$$

The ultimate result, when all substitutions are made is Equation 34:

$$R'\frac{Q_{n+1} - Q_n}{\Delta t} + \frac{Q_{n+1}}{C'} = \frac{V'_{n+1} + V_n}{2} + \frac{Q_n}{C_0} - \frac{V_{cat}(t_{n+1}) + V_{cat}(t_n)}{2}$$

References

[1] F. Chen, *Introduction to Plasma Physics, first edition*, Chapter 9, Plenum Press 1974

[2] F. Chen, *Introduction to Plasma Physics, first edition*, page 87, Plenum Press 1974

[3] J. R. Pierce, *J. Appl. Phys.* **15**(1944), 721

[4] C. K. Birdall and W. B. Bridges, *Electron Dynamics of Diode Regions*, Chapter 3, Academic Press, 1966

[5] P. Burger, *J. Appl. Phys.* **36**(1965), 1938

[6] P. Burger, D. A. Dunn and A. S. Halsted, *Phys. Fluids* **8** (1965), 2263

[7] T. L. Crystal and S. Kuhn, *Phys. Fluids* **28**(1985), 2116

[8] S. Kuhn, C. K. Birdsall, T. L. Crystal, P. G. Gray, and Wm. S. Lawson, *1984 International Conference on Plasma Physics, Proceedings, Contributed Papers, p. 104 — Lausanne, Switzerland.* (This paper has a movie associated with it.)

[9] K. Y. Kim and T. L. Crystal, *Proceedings of the Second (1984) Symposium on Plasma Double Layers and Related Topics*, page 290. Organized by: Plasma Physics Group, Institute for Theoretical Physics, University of Innsbruck, A-6020 Innsbruck, Austria

[10] N. F. Otani, *Proceedings of the Second (1984) Symposium on Plasma Double Layers and Related Topics*, page 296

[11] Wm. S. Lawson, "PDW1 User's Manual", University of California / Electronics Research Laboratory Report M84/37, 27 April 1984

[12] C. K. Birdsall and A. B. Langdon, *Plasma Physics Via Computer Simulation*, McGraw-Hill Book Company, 1985

[13] R. W. Hockney, *Methods in Computational Physics* **9** (1970), 136

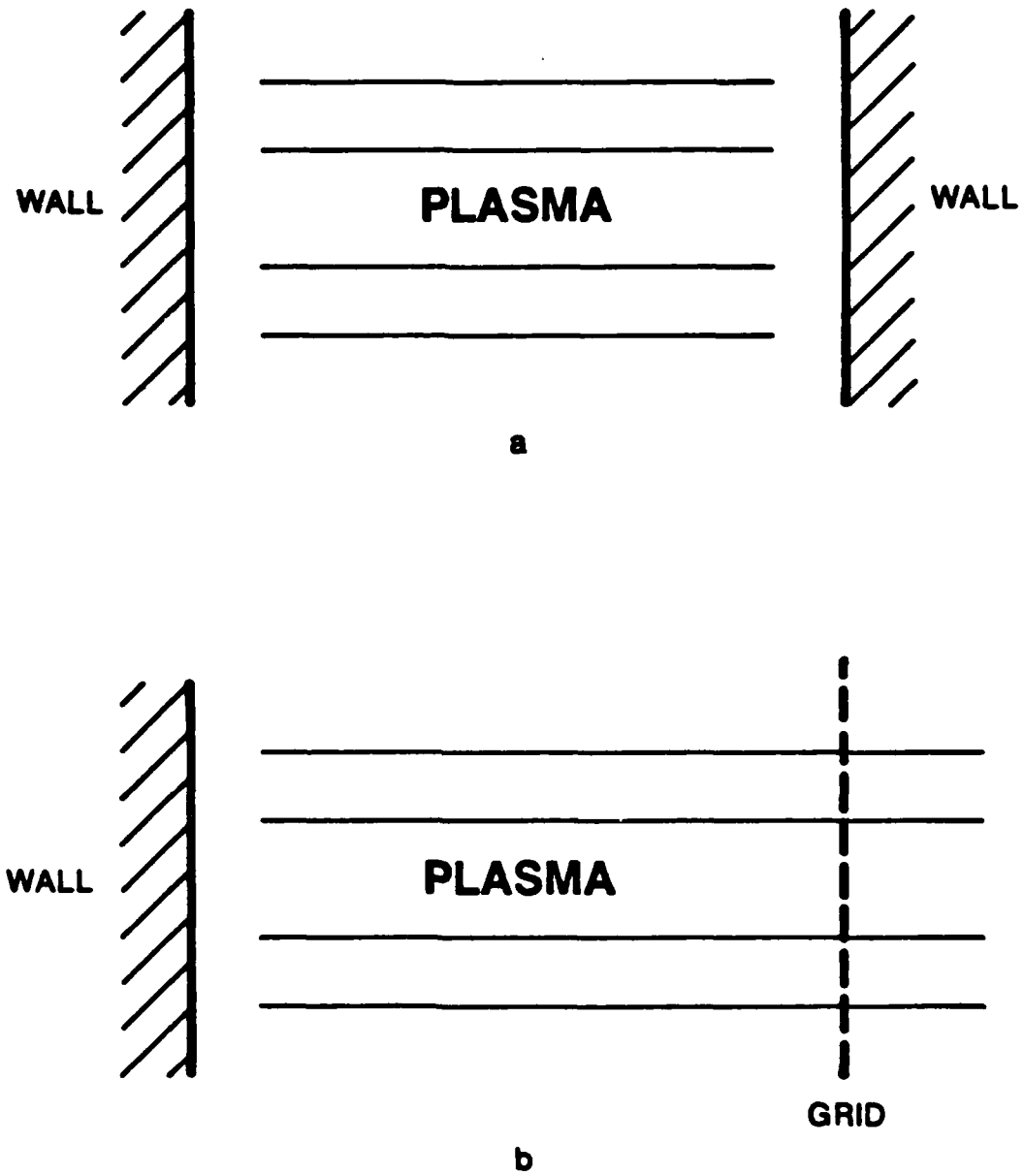


Fig. 1. Different 1-d models can have identical solutions: (a) is contained between two walls, (b) is open-ended, but region of interest is only between grid and wall. As long as the particles which enter the region of interest are exactly the same in both cases, (a) and (b) will yield identical results.

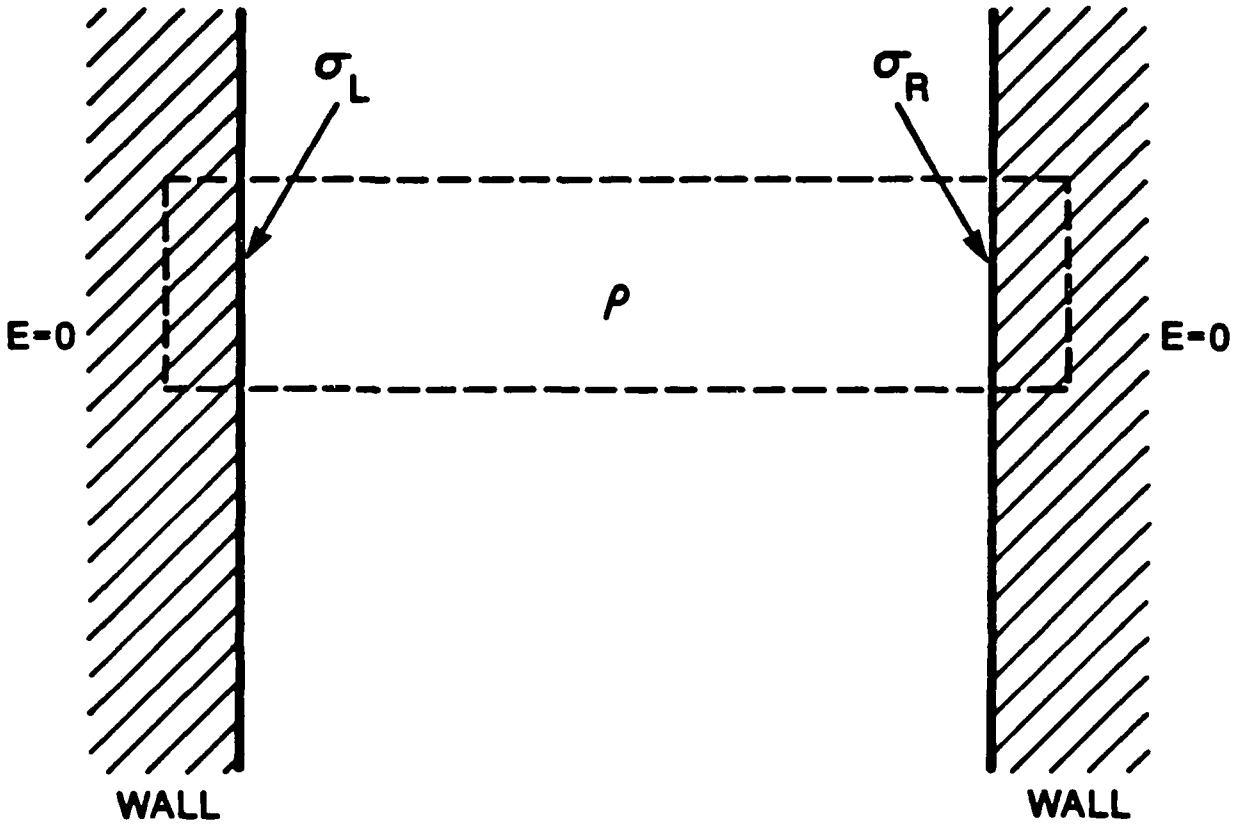


Fig. 2. According to Gauss' law, $\sigma_L + \int \rho dx + \sigma_R = 0$.

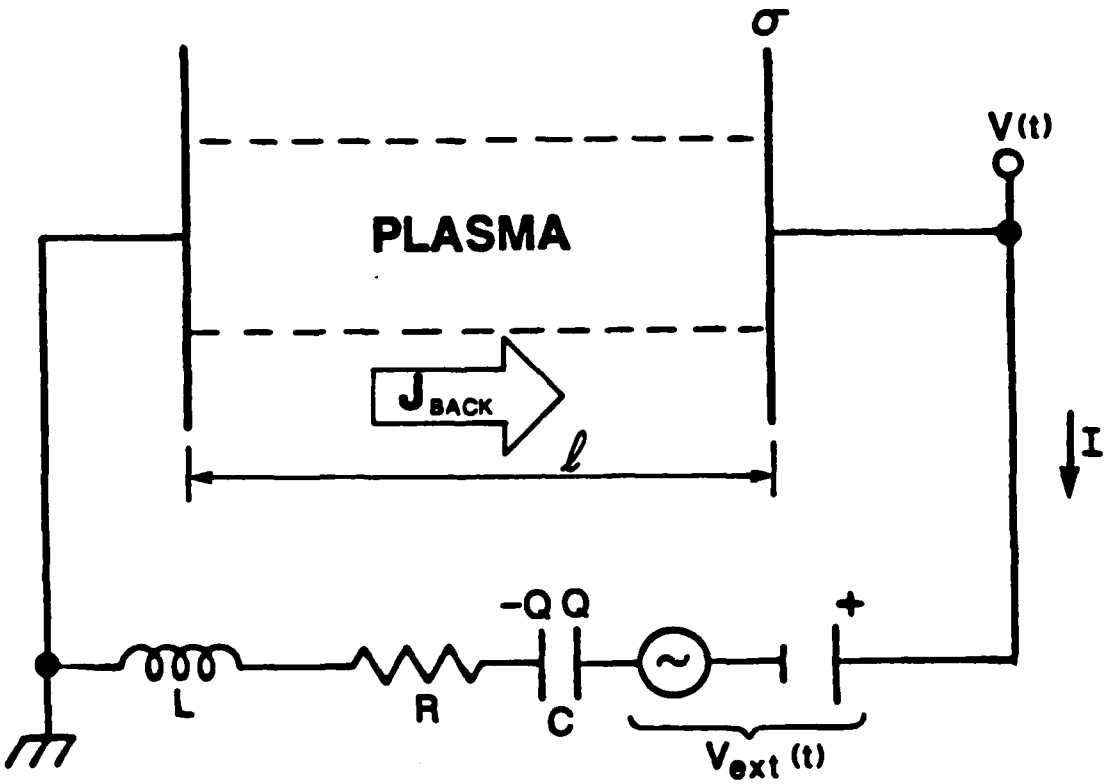


Fig. 3. Model for external circuit with dynamical variables σ, V, I, and Q as defined in text. Arrow represents a constant background current density (which could also be represented as an external constant current source in parallel with the external circuit shown).

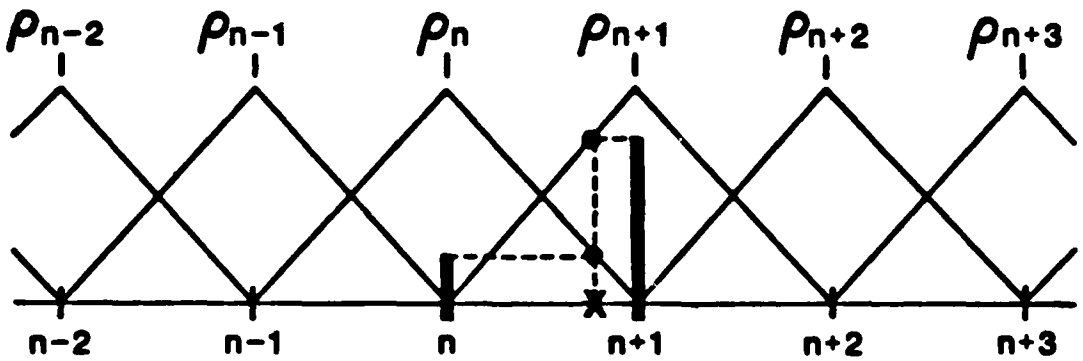


Fig. 4. Assignment of particle charge to grid — linear weighting. The charge assigned to the grid points due to the particle at X is represented by the heavy vertical lines.

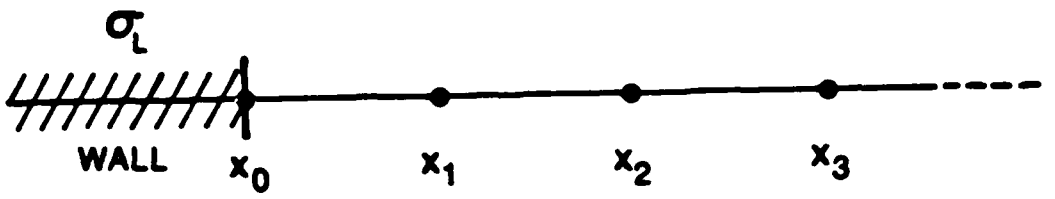


Fig. 5. Locations of grid points relative to boundary.

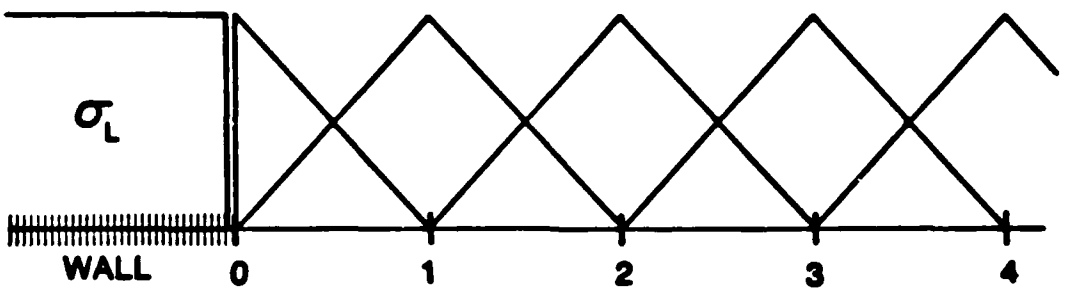
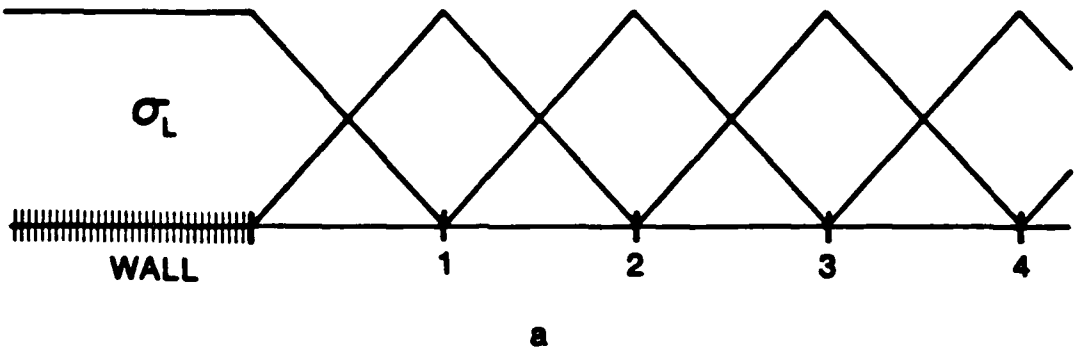
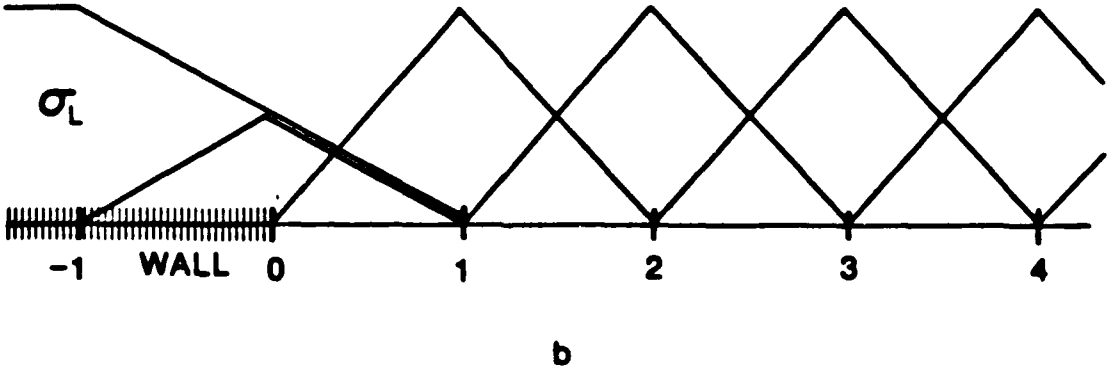


Fig. 6. Linear weighting for a "hard" boundary system. Note the triangle for the grid point at the boundary is cut in half by the boundary, so that only half as much charge accumulates there as on nearby grid points.



a



b

Fig. 7. Linear weighting for two "soft" boundary systems: (a) lumps together charge on and adjacent to the boundary, and (b) weights them smoothly.

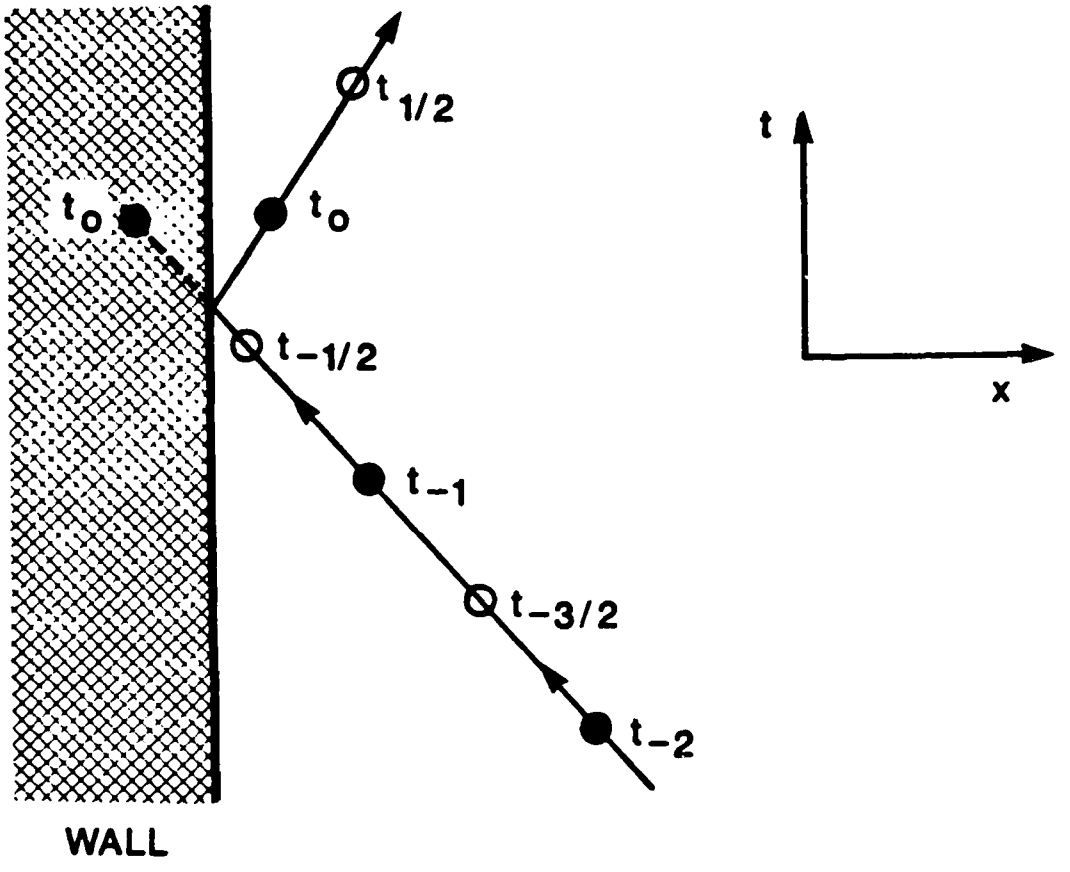


Fig. 8. Positions are known at solid dots, velocities at hollow dots.

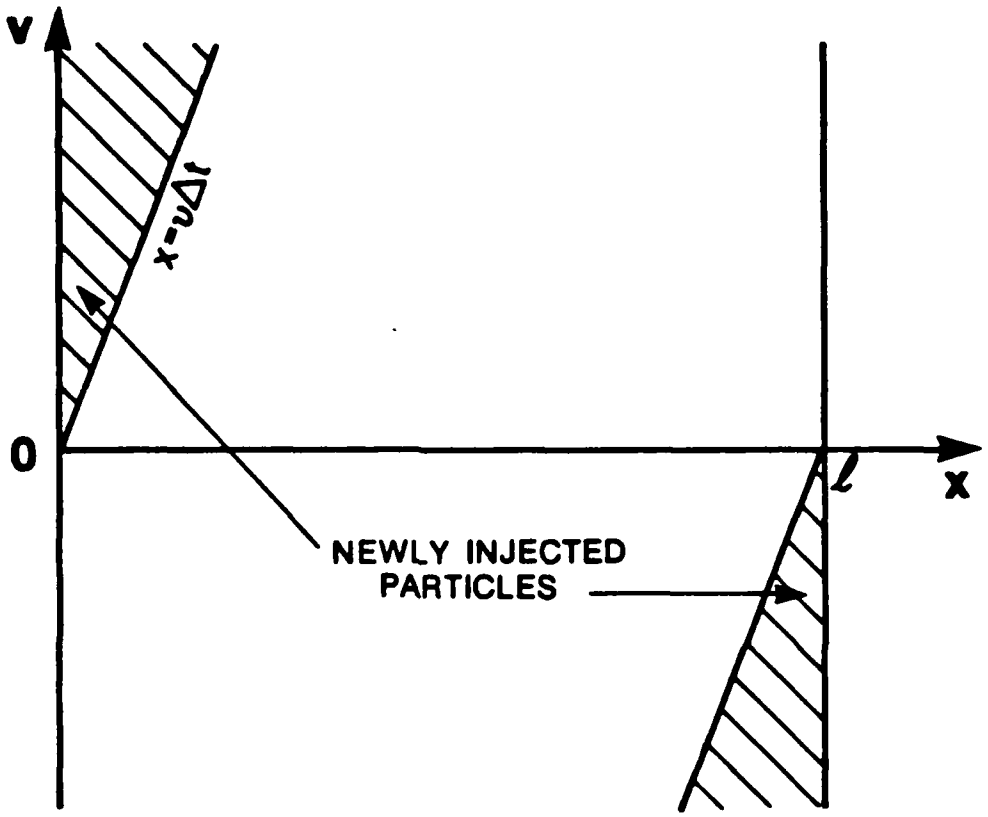


Fig. 9. Region of phase space which must be filled with new particles each time step.

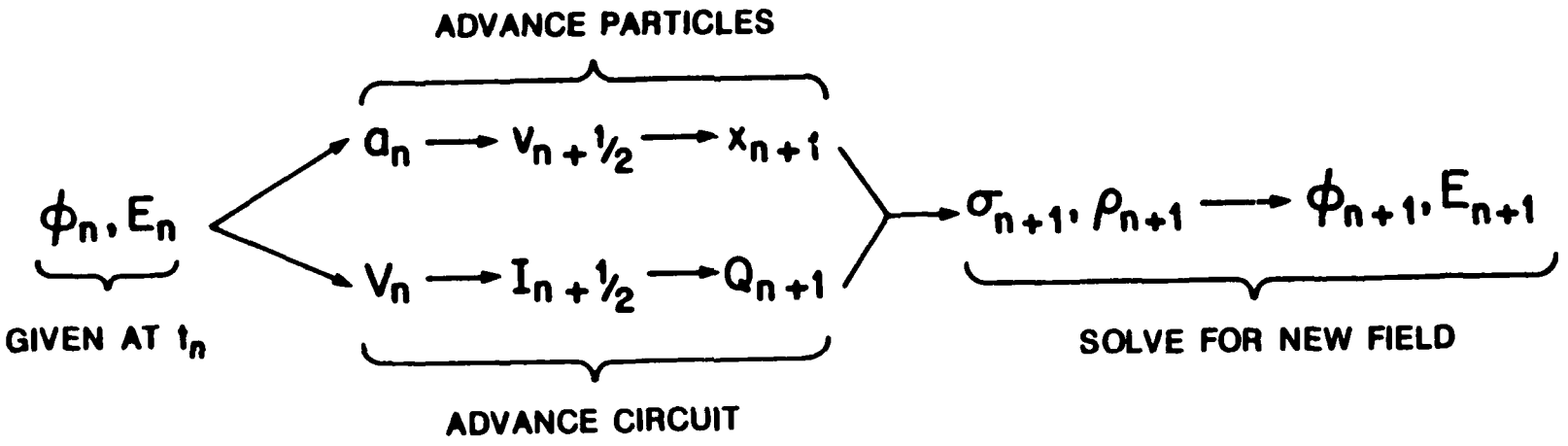


Fig. 10. Scheme for advancing simulation one time step when $L \neq 0$. Particles and circuit are advanced simultaneously.

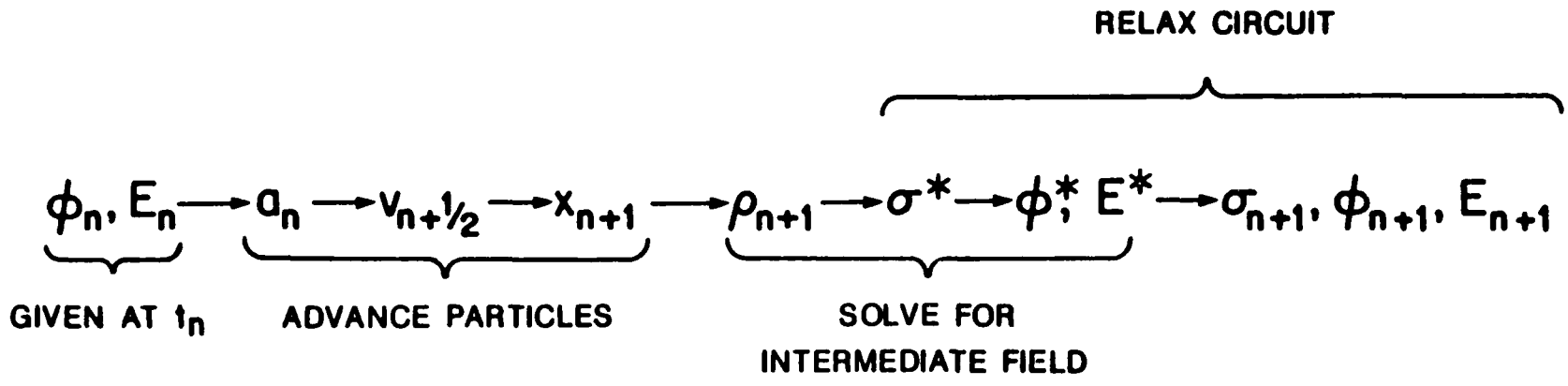


Fig. 11. Scheme for advancing simulation one time step when $L = 0$. Particles are moved first, then circuit is relaxed.

END

6-87

DTIC